Systematic Construction of Nonlinear Product Attacks on Block Ciphers

Nicolas T. Courtois[1], Matteo Abbondati[2], Hamy Ratoanina[1], and Marek Grajek[3]

[1] University College London, Gower Street, London, UK
[2] Independent maths teacher based in London, UK
[3] Independent cryptography and crypto history expert, Poland

Abstract. A major open problem in block cipher cryptanalysis is discovery of new invariant properties of complex type. Recent papers show that this can be achieved for SCREAM, Midori64, MANTIS-4, T-310 or for DES with modified S-boxes. Until now such attacks are hard to find and seem to happen by some sort of incredible coincidence. In this paper we abstract the attack from any particular block cipher. We study these attacks in terms of transformations on multivariate polynomials. We shall demonstrate how numerous variables including key variables may sometimes be eliminated and at the end two very complex Boolean polynomials will become equal. We present a general construction of an attack where multiply all the polynomials lying on one or several cycles. Then under suitable conditions the non-linear functions involved will be eliminated totally. We obtain a periodic invariant property holding for any number of rounds. A major difficulty with invariant attacks is that they typically work only for some keys. In T-310 our attack works for any key and also in spite of the presence of round constants.

Key Words: block ciphers, Boolean functions, Feistel ciphers, weak keys, DES, Generalized Linear Cryptanalysis, polynomial invariants, multivariate polynomials, annihilator space, algebraic cryptanalysis, polynomial rings, invariant theory.

1 Introduction

Block ciphers are widely used and studied since the 1970s. Their periodic structure is prone to round invariant attacks, for example in Linear Cryptanalysis (LC). A natural generalisation is Generalised Linear Cryptanalysis (GLC), first proposed at Eurocrypt'95 [29]. The space for possible attacks grows double-exponentially, and until 2018 extremely few such attacks [30, 19, 37, 17, 4] have been found. We call a "product attack" an attack, where an invariant, being a product of simpler polynomials, remains unchanged after some number of $k \geq 1$ rounds. A key point is that in the ring of Boolean polynomials the factorization is not unique. This has important consequences. Numerous specific events without unique factorisation occur inside many invariant attacks, cf. [11], making the job of the attacker easier. Then, imagine that a researcher finds a new invariant attack which works for a block cipher. It could be very difficult to know if this attack can or not be constructed by multiplying some well chosen polynomials as in our general "product" attack framework which we introduce in this paper.

An essential question is whether invariant attacks do exist at all for any given cipher. This question is currently considered very difficult [3, 4, 6]. For many ciphers we can neither say if it is broken by our attack, nor we can be assured that it is secure and invariant attacks do not exist. Numerous positive examples of working attacks are known for the Cold War cipher T-310 [35, 25]. There exist also some basic examples for DES [19, 18] which we will revisit here. Then we have results on SCREAM, iSCREAM, Midori64 and MANTIS-4 cf. [37, 4]. Most previous non-linear attacks exploited polynomials of degree 2 or 3 [19, 37, 17] and only sometimes of higher degree [17, 18], or the invariants are only correct with a low probability. In this paper we construct invariants of arbitrarily high degree and working with probability 1, in a systematic deterministic way.

This paper is organised as follows. In Section 2 we explain what are non-linear invariant attacks and key features of our approach. In Section 3 we explain the idea of "closed loop" connection. In Section 4 we describe the main idea of cycles with transitions between polynomials. In Section 5 we discuss the question of attacks working with strong rather than weak Boolean functions. In Section 6 we present a simple attack at degree 4. In Section 7 we present our general framework theorem. In Section 8 we apply it to construct a stronger attack of with a cycle of length 8. In Section 9 we apply our construction to DES with 2 cycles of length 8. In Appendix A we give two different mathematical proofs that our complex attack on DES, actually works. In Section 10 we provide a better attack on DES at degree 5. In Appendix B we consider DES with original S-boxes.

2 Our Methodology, Scope, Applicability, Features

We call $\mathcal{P}$ a polynomial invariant if the value of $\mathcal{P}$ is preserved after one round of encryption, i.e. if $\mathcal{P}(\text{Inputs}) = \mathcal{P}(\text{Outputs})$. This concept can be applied to any block cipher except that such attacks are quite hard to find, cf. [3].

In this paper we introduce a general method for constructing polynomial invariants of high degree designed to work on more than just one well chosen cipher configuration. Moreover our attacks do NOT seem[1] to require that a block cipher has any special property or weakness. We only use properties, which are very common and which essentially any block cipher ever made has. We assume that our cipher includes a sequence of applications of non-linear functions which transforms the state bit by bit, and different polynomials on the state are constructed step by step, without the necessity of knowing how the whole state is computed. Moreover we relax these transitions in the strongest possible way: some transitions assume that they actually do not hold at all, or more precisely their difference is assumed to be an arbitrary non-linear function of the cipher state (which we will later try to eliminate algebraically).

Many research papers in symmetric cryptanalysis spend a lot of time studying the specification of a given block cipher. In this paper we emphasise the idea that it is not necessary to know the full specs of a cipher in order to find an invariant

[1] Except maybe some combinatorial or probability questions for certain special events.

attack. For T-310 we make an essential and deliberate choice of not providing the full specs which are excessively complex, cf. [23]. For DES we assume that the reader is familiar with the basic description of DES. The purpose of this is threefold. First, we want to demonstrate that by their very nature our attacks represent self-contained mathematical results about polynomials involving very few variables, which are **able to eliminate** everything else. Secondly, and as such, our attacks will apply to potentially any cipher, which after renaming the variables satisfies the same basic set of polynomial relations, which will be organised in order to form short cycles, as we will see later. Finally, we want to emphasise the fact, that our attack depends only on a tiny fragment [2] of the cipher's computational circuit. Many traditional attacks depend on probabilistic events on the cipher state. Therefore by their very nature they require to know the full specs of the cipher, in order to know if they work as expected. Sometimes they don't work as predicted, because certain events are biased or not independent. Here polynomial attacks are different: they are theorems on combinations of Boolean polynomials and on relations between different bits holding always, with probability 1, under the conditions specified, for any cipher input. There is no need to be able to compute the cipher circuit in its entirety in order to validate them. For T-310 there are simply no special cases where our attacks would not work as predicted.

For DES we formulate our results in such a way, that the key bits are included inside the S-box specification. Our results, such as later Thm. 10.1, do not make an apparent reference to the secret key. Or rather this question needs to be studied separately[3] when our invariant would be applied inside some attack, cf. Section 9 in [18] and Section 6 in [13]. In general our attacks are meant to be existential over the secret key: work for a fraction of key space which should be as large[4] as possible.

2.1 Limitations and Vulnerability

Some polynomial invariant attacks work for a fraction of keys, other for all possible keys. According to [17, 18, 26], using longer keys in each round could be a good reason, why many ciphers are likely to be secure against non-linear attacks. However even when many key bits are used in each round, cf. [25], it is hard to be sure that a cipher is not vulnerable to the same type of attack. The crucial notion here is the diffusion and the combinatorial problem of the existence of "closed-loop" sub-circuits which was emphasised recently in [38]. This type of property was already studied long time ago, cf. for example Fig.

[2] Involving a handful of bits, and only some of the non-linear function(s), and only some key bits. Moreover inside the Boolean functions and S-boxes, we aim at constructing attacks which require only that a certain a small fraction of entries in the truth tables of these functions (at suitable positions) are at zero.

[3] For DES S-boxes, we require that some Boolean function are annihilated by products of simple linear polynomials. Such annihilation remain frequently true, when we transform an S-box by a secret key, added at the input, cf. Remark 2 in page 22.

[4] In T-310 (but not with DES) our attacks work for any key and also in presence of round constants, e.g. in Thm. 6.2 and numerous others examples in [17, 18].

3-5 in [22], Section 9 in [14] and Fig. 10 in page 21. The philosophy is that our attack is facilitated, if some subset of bits depend "mostly" on themselves, and only "weakly" on other bits inside one round of the cipher. More precisely all the other bits need to be eliminated by polynomial algebra. In this paper we will take this idea to a new level, cf. Section 3, and allow linear combinations of bits.

2.2 What is New - On Existence of Cycles on Basic Polynomials

Can we do better than current (heuristic) approaches? This paper introduces a substantially and **strictly more general** paradigm, which increases the number of possibilities for the attacker. This hopefully leads to more (or better) attacks on block ciphers. Instead of looking at bits, and how they depend on other bits, we will actually ignore individual bits and considerably restrict the set of values which we actually need to study. We consider ONLY a certain (small) set of "basic" polynomials $\mathcal{Q}_i$ involving these bits. Then we consider cycles built from such linear/affine [or more generally non-linear] polynomials. Is this possible? Cipher designers have 50+ years of experience in designing complex ciphers aiming at avoiding such attacks. We expect that in most cases no cycles whatsoever involving polynomials of "tractable" size will be found.

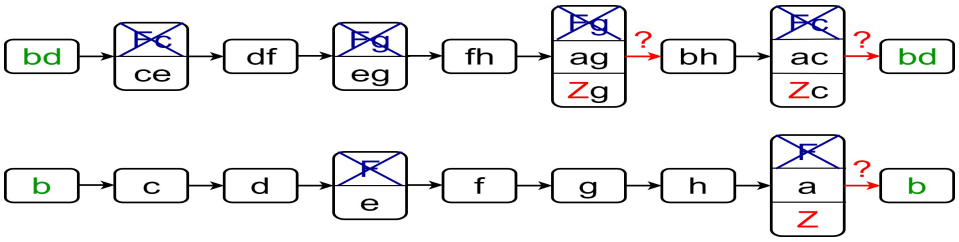

Fig. 1. Example showing how simple polynomials, e.g. bd, are transformed and maybe eventually form cycles in T-310, cf. Section 7.4. in [17]. Terms with crosses in blue such as Fc appear an even number of times and are eliminated mod 2. Terms and transitions with Z in red work, if certain conditions on the Boolean function Z hold. This paper is about how such examples can be constructed from scratch in a systematic way.

Therefore it is crucial to be able to enhance this basic approach in order to increase the number of possibilities for the attacker. We make the "impossible" question of the existence of short cycles eventually possible. In order to achieve this we cheat in some way and study **imperfect** transitions, cf. also later Section 4 and 3. Certain arbitrary non-linear functions $\mathcal{Z}_i$ are added on the way. Then eventually we eliminate these extra functions $\mathcal{Z}_i$ algebraically [annihilation of polynomials], which is the main idea which eventually makes our attacks work.

2.3 Related Research, Product and Multiple Product Attacks

In recent research there are two major types of invariant attacks: linear or affine sub-space invariants [31, 3, 6], and more generally, arbitrary non-linear polynomial invariants [37, 17]. Several authors [3, 6, 17, 4] study both topics which are closely related. In some cases both paradigms are simply equivalent. For example we consider the concept of the so called "product attack" of [18]. It is easy to

see that any affine vector sub-space of $\{0,1\}^n$ can be also described as a set of points, where a product of some affine polynomials $\mathcal{Q}_i$ is at 1. Likewise every set of points in $\{0,1\}^n$ where $\prod \mathcal{Q}_i = 1$ is an affine subspace. Then, linear spaces can be characterised by restricting to the case of linear polynomials $\mathcal{Q}_i$.

In this paper we aim at improving and generalizing such "product" attacks. It is important to see that an attack with a sum of several products will be more general. A somewhat misleading example can be found in Appendix A.2. of [18]: it is a sum of two products however a close examination would show it can also be written as a single product (!). A better example can be found in Section 10.6 of [17], where the invariant is of type $AC + BD$ where A, B, C, D are linear polynomials and $AC + BD$ is irreducible. Different attacks are related to each other, and the exact invariant of type $AC + BD$ in Section 10.6 of [17], hides the existence of a "product" attack for 2 rounds of type $AC \to BD \to AC$. This demonstrates that the product attack is NOT necessarily the most general attack. In general it is easy to show that the set of all possible invariants is a polynomial ring; both addition and multiplication are allowed (!). Then the construction of this paper, which emphasises the multiplication, is just the first step. This paper essentially aims at solving the problem of the invariant ring being not empty [existence of at least one invariant attack]. Then, our experience shows that frequently the ring of invariants, will contain additional[5] lower degree invariants, not anticipated[6] from the initial product attack.

2.4 Are Polynomial Invariant Attacks on Block Ciphers Possible?

We are looking for a polynomial which is preserved when we apply one round of encryption, i.e. if $\mathcal{P}(\text{Inputs}) = \mathcal{P}(\text{Outputs})$. Now in finite fields any function is a polynomial and the outputs are also polynomials in the inputs. If we substitute these inside $\mathcal{P}$, we obtain another polynomial, initially very complex, however if $\mathcal{P}$ is well chosen, it will be simpler than expected. In particular the key bits can be eliminated and after this, in our you polynomial was an invariant, we get a situation where, two polynomials[7] will be simply equal. We get a formal equality on two complex polynomials, holding under certain constraints on the key or/and cipher wiring. Interestingly, if $\mathcal{P}$ is a product it is easy to see, that both polynomials are products of many terms. It appears that when two products of polynomials are equal, this does not happen by accident. The ring of Boolean polynomials does not have unique factorisation, and we observe specific types of events: annihilation events, absorption events, etc, cf. [11]. In this paper, for the first time ever, we are going to abstract the non-linear invariant attacks from any block cipher in particular. We are going to formulate our attacks in such a way, that they do not depend on features of any particular block cipher. What

[5] For example, we have generated many concrete examples of S-boxes, for the ttack of degree 8 on DES of Section 8. In some cases additional invariants of degree 2,4,6 or 7 are also found, cf. [12, 11] and our Section 8. Or we constructed an attack of degree 10 in Section 9 and for some S-boxes we will also have an attack of degree 5 in Thm. 10.1.

[6] Some of these attacks are obtained by so called "decimated" attack cf. Section 4.3

[7] The original one and the transformed one.

we do amounts to doing a "clever" polynomial algebraic combination of a few equations, which are basic facts about how some polynomials are transformed by our cipher.

2.5 Related Work: Linearization, XL, Algebraic Cryptanalysis

Our new attack could be called "Product Cycling Linearization Attack" on block ciphers and is vaguely related to other works which use the word "Linearization" cf. [9, 32]. The main idea with linearization (in all cases) is to add new variables so that everything becomes linear and then try to eliminate these new variables. In XL and old "Linearization" [9] we multiplied complex non-linear equations by various variables. In algebraic attacks on stream ciphers [15] we multiply them by well chosen non-linear polynomials. In this paper we multiply non-linear functions by well chosen polynomials which are products of linear factors.

3 Closed Loop Configurations Revisited

In recent research [38, 21, 22, 17, 18] it turns out that each time an "interesting" non-linear invariant attack was found, it comes together with a configuration where some set of bits and S-boxes are primarily connected to each other in a "closed-loop" cf. [38]. This idea is not new, for instance for T-310 it was studied in Sections 7.4, 8.2, 9.1. and 9.2. and 10.6 and 10.7. in [17]. Such configurations occur in almost every attack previously studied. For example we may look at Fig. 8 in [17] which is reproduced below as Fig. 2. Likewise Fig. 7 in [17] is reproduced in page 8 as Fig. 4. We can also mention Fig. 9,10,11,12 in [17].

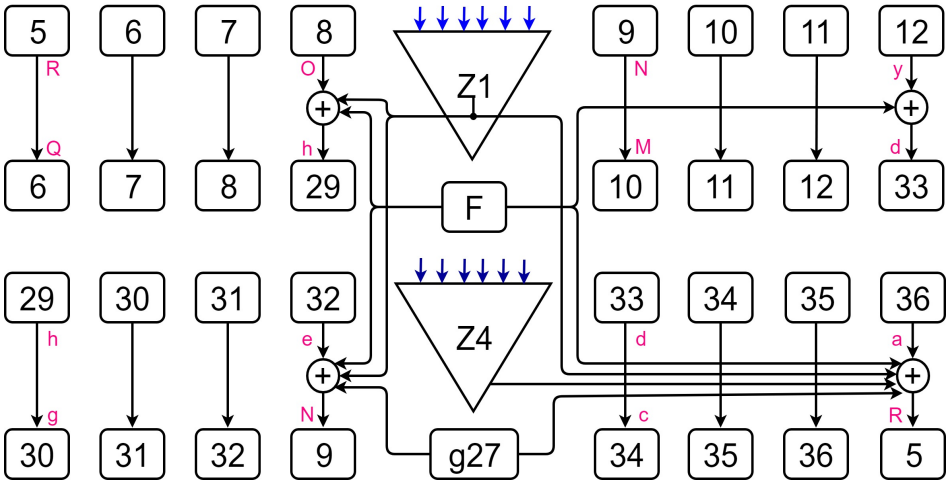

Fig. 2. A complex closed loop configuration for T-310 with 16 active bits.

For GOST this type of configuration was already studied in Fig. 3-5 in [22] and GOST is known to be a particularly weak cipher in this aspect which is closely related to vulnerability of GOST to truncated differential attacks cf. [10, 21, 22, 31].

Closed loop configurations can be of any size. On Fig. 10 page 21 we show how this works with DES S-boxes 2,3,7 and a well chosen set of inputs and outputs of these bits. A larger configuration with five S-boxes is shown in Fig. 3.

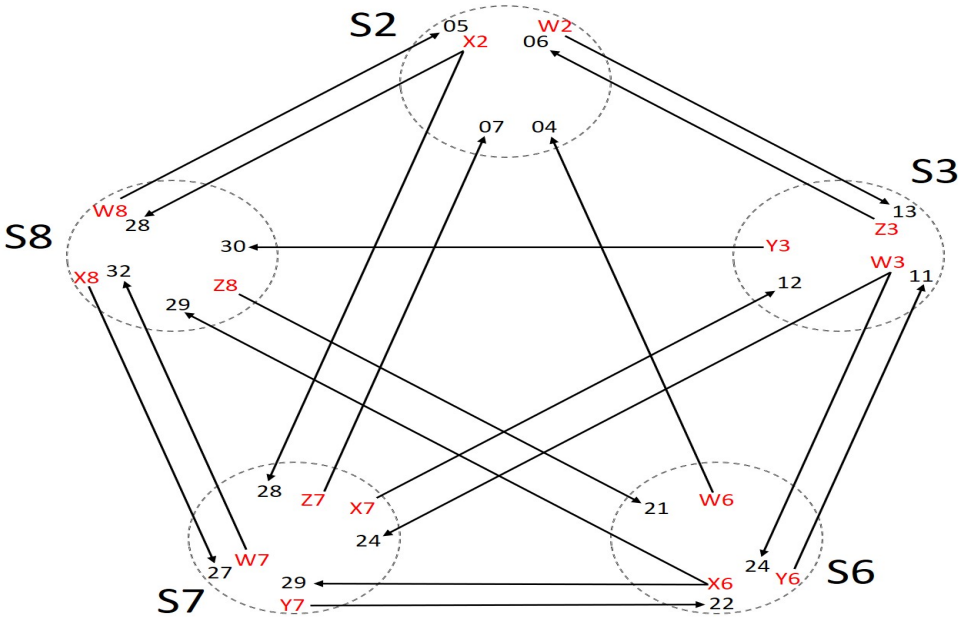

Fig. 3. Closed-loop connection between S-boxes S2,S3,S6,S7,S8 in DES.

This configuration is one of the best possible in DES, it maximises the number of bits active across all possible subsets of 5 S-boxes. Intuitively, the more bits depend only on themselves, in terms of a ratio (or probability) of re-entry, the more we can hope to find an invariant attack.

In this paper we take this idea to a new level: we consider not bits, but rather their linear [and also non-linear] combinations. Intuitively, it seems unthinkable that any block cipher can be cryptanalyzed by showing that it acts in a simple way while acting on some small set of "simple" polynomials and that we can ever obtain short cycles in this way, cf. also Section 2.2. For this reason, our cycles are going to be **imperfect** in the following sense: we allow **addition** of several arbitrarily complex non-linear functions which are later eliminated. This is expected to increase the number of possibilities for the attacker.

4 Constructing Cycles on Polynomials

Only those who attempt the absurd will achieve the impossible.
– Maurits Cornelis Escher

We are now going to imagine a larger enhanced graph, where edges represent some polynomials $\mathcal{Q}_i$, and transitions correspond to how these polynomials might

be transformed by one round of encryption, if certain conditions are true. Let $A, B, \ldots$ be some linear combinations of input bits for one round. By convention we say that $A \to B$ if $A(\text{Inputs}) = B(\text{Outputs})$ each time we look at one encryption round. If this happens for every key and for every input we say that $A \to B$ holds without any condition. More generally in our construction we will have basic $\mathcal{Q}_i$ being affine combinations of $A, B, \ldots$, which in turn are well-chosen polynomials. These basic polynomials $\mathcal{Q}_i$ and $A, B, \ldots$ are usually linear or affine. For example $\mathcal{Q}_1 = A+C+1$, where A and C are two affine expressions[8] in cipher state input variables, and addition is done modulo 2. In general they are meant to form short cycles holding under certain technical conditions.

For example in a hypothetical attack we could have a cycle, such as say $A \to B \to C \to D \to A$, which does NOT work as such. Initially some transitions are just impossible, in particular $D \to A$ does not work. Interestingly, we can apply the following idea borrowed from [12] and Section 5 of [17]. Let $\mathcal{Z}_i = 0$ be a transition polynomial. Informally, the transition polynomial $\mathcal{Z}_i = 0$ characterises exactly the cases where this transition actually works[9].

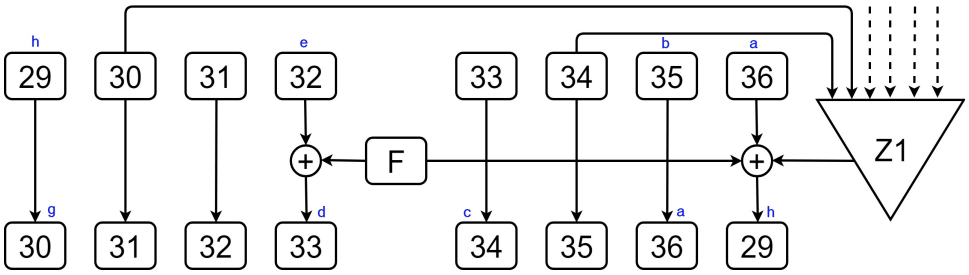

Fig. 4. Transitions inside one simple invariant attack on T-310, cf. [17].

Here is another example, where we attempt to construct a cycle of length 4 which is illustrated in Fig. 4. We could for example have $D = x_{32} + x_{36}$ and $A = x_{29} + x_{33}$, $B = x_{30} + x_{34}$ and $C = x_{31} + x_{35}$, where x_i are inputs of one round of encryption, and $Z1(x_1, \ldots)$ is some specific non-linear function. Then we imagine that the round constant F is[10] eliminated but the output of $Z1()$ is not eliminated and we have $D \to A$ only when $Z1(\text{Inputs}) = 0$. Importantly we are not going to assume that $Z1()$ will be equal to 0 for every input. We expect that the Boolean function $Z1$ is quite strong.

[8] This example occurs in our later attack on DES, and we can rewrite $\mathcal{Q}_1 = A+C+1 = R05 + R28$, where $R05$ is the 5-th bit in the right branch of a DES plaintext, and A, C are defined in later Fig. 11 page 27.

[9] Informally it is a polynomial such that we actually have $D \to A$ when $\mathcal{Z}_i = 0$. Moreover we mandate that this polynomial $\mathcal{Z}_i$ uses the same set of input-side variables which are also the inputs of D. Then we always have $D(\text{Inputs}) = A(\text{Outputs})$ when $\mathcal{Z}_i(\text{Inputs}) = 0$. This does not say what happens when $\mathcal{Z}_i = 1$, and in this paper the converse will also hold systematically. More precise statements which make sense in all cases will be provided later, cf. Thm 7.1 page 16.

[10] In T-310 cipher F is derived from the public IV used in each encryption, cf. [23].

We will denote each such transition polynomial by $\mathcal{Z}_i$ pertaining to a transition number i. It generalizes the concept of Fundamental Equation (FE) of [17] to arbitrary[11] transitions[12]. The main idea in this paper is that following [12] in many cases this polynomial is not going to be zero. However some multiple of this polynomial is more likely to be zero.

We aim at constructing a configuration with one or several cycles, which can be seen as walks in some graph where our block cipher is acting in a non-deterministic way on a set of our basic polynomials. In addition in some cycles, non-linear polynomials $\mathcal{Z}_i$ are added on the way, in the same way as $Z1()$ in Fig. 4 above, or with $Z \cdot g$ term in earlier Fig. 1. These transition polynomials, initially seem to be a huge obstacle in constructing our attack. Eventually we will show that under suitable annihilation events, $\mathcal{P} = \prod \mathcal{Q}_i$ is an invariant for our cipher, cf. our Thm. 7.1 page 16.

4.1 Non Deterministic Walks On Cycles

We operate by "walks", advancing by one position on a given cycle on polynomials $\mathcal{Q}_i$. This process is not deterministic and the path is not unique. The same polynomial $\mathcal{Q}_i$ could potentially appear on several cycles. This is due to some freedom[13] of choice for the $\mathcal{Z}_i$. This greatly increases the number of possibilities for the attacker. In this paper, for the sake of simplicity, all cycle walks are deterministic.

4.2 Discussion, Success Probability

The attacker works with arbitrary sets of well-chosen cycles. In the basic version of our attack, we simply multiply all the $\mathcal{Q}_i$, and we expect to get a non-zero polynomial invariant, which can[14] then be used in cryptanalysis. Can this be made to work? One factor which increases the chance of success is the size of our configuration with all the $\{\mathcal{Q}_i; \mathcal{Z}_i\}$. The more polynomials we multiply, the more likely it happens that all our polynomials $\mathcal{Z}_i$ are annihilated by some product of the $\mathcal{Q}_i$. It is easy to see that if a [Boolean] polynomial g is a product then for a Boolean function f to be annihilated by this polynomial, i.e. $fg = 0$ for every input, we just need to look at values of f at points where $g = 1$. This will concern only a fraction of the truth table of f, and therefore is more likely to happen. Specific examples are provided later. In Section 6.3, g is linear, which makes the attack very weak, cf. later Thm. 6.4. Then in an improved attack in Thm. 8.1 in Section 8 every g has 2 factors. Similarly for DES, in [18] some g are linear, then in Thm. 9.1 in Section 9 each of five polynomials g has two factors.

---

[11] This type of equation was previously studied under the name of a *Transition Equation* or (TE) in Section 5 of [17].

[12] Here transitions are no longer invariants but rather of type $\mathcal{P} \to \mathcal{P}'$ with $\mathcal{P} \neq \mathcal{P}'$.

[13] It is easy to see that there is no reason why transition should be deterministic. For example we could have $\mathcal{Z}_1 = Z(a, b, c) = abc + ac$ and $\mathcal{Z}_3 = Z(a, b, c) + b = abc + ac + b$ which inevitably lead to two different transitions if starting form the same polynomial assuming $\mathcal{Q}_1 = \mathcal{Q}_3$, and we have simultaneously $Z(b+1)(a+1) = 0$ and $(Z + b)(b + 1)(a + 1) = 0$.

[14] We refer to Section 9 in [18] and Section 6 in [13] to see how.

Our attacks work only for as long as $\mathcal{P} = \prod \mathcal{Q}_i$ remains not zero itself, which strongly limits what we can achieve.

4.3 Additional Attacks with Decimation and Sub-Cycles

Decimated variants based on sub-cycles can also be constructed. The main idea is that we can advance by more than 1 step in a cycle. This happens for example in T-310, under certain (more stringent) technical conditions we can have an attack with period of 2 rounds[15] of type $AC \to BD \to AC$ cf. [12], which uses the same cycle of period 4, which will also be used in our attack in Fig. 7. For DES we provide in Section 10 an example of type $\mathcal{P} \to \mathcal{P}' \to \mathcal{P}$, where the degree of $\mathcal{P}$ and $\mathcal{P}'$ is 5. Then it is easy to see that $AC + BD$ and $\mathcal{P} + \mathcal{P}'$ are invariants holding for[16] 1 round.

5 Limitations vs. Vulnerable Boolean Functions

Main limitations of our attacks are that the degree of $\mathcal{P}$ in the attack must increase [17] substantially in order to work with some non-trivial (e.g. highly non-linear) Boolean functions or S-boxes. Then potentially some attacks at lower degree can be also constructed as shown in Section 10. However, typically there is a price to pay for such improved attacks to work at a lower degree[18]. For example, our advanced degree 5 result of Thm. 10.1 has a serious drawback. It forces the attacker to use some annihilator g, which is linear, making that this second attack works in extremely few cases, cf. Thm. 6.4. The same problem occurs in the attack on Section 11.7 in [18].

5.1 On Annihilation Vulnerability and Worst-Case Normality

The primary aim of this paper is to have an even more general attack framework, leading to annihilation assumptions which are more likely to work also when Boolean functions are strong. Our long term goal is to find attacks such as in [18], which operate under weaker and fully realistic assumptions. For example such that every annihilator g is a product of three affine factors, e.g. with $Z(a+b)c(1+e) = 0$. One question is if there exist non-linear attacks using such properties at degree 3. For example following [17, 18] for DES this is already quite difficult to achieve [19] A second question is how many boolean function are vulnerable. There are very few examples in real-life encryption systems, where Boolean functions would have annihilators of degree 2. In contrast properties

[15] Examples of non-linear invariants with a period of 4 rounds can be found in Appendix B.2. in [17].

[16] In contrast, due to the lack on unique factorisation in product attacks, it is not clear if or how our attack of degree 5 in Section 10 can be obtained, with or without decimation, from cycles following our general framework.

[17] This is related to the question of biases inside the block cipher induced by polynomial invariants, cf. Section 9 in [18] and Section 6 in [13].

[18] Rather than when we simply multiply all the polynomials.

[19] The best example known to us so far requires $\mathcal{P}$ of degree 20.

with degree 3 annihilators can very hardly be avoided in general, which we are going to show now.

One recent and surprising result is that the probability that for example $Z(a+b)(c+d)(e+f)=0$ is typically quite high, cf. [18]. Accordingly if this property does not hold for the original Boolean function of T-310, this is maybe just accidental rather than deliberate. For example $Z(a+b)c(1+e)=0$ holds for the original Boolean function designed in the 1970s, cf. Appendix I.19 in [23]. For general cubic annihilators the Thm 6.0.1 in [15] says that for every Boolean function Z we have either Z or $Z+1$ which has an annihilator of degree 3. Then Thm. 6.3. and Thm. 6.4. in [18] deals with special cases which split into some specific affine factors.

This property is very highly relevant to our general framework and all attacks studied in this paper. It is not immediately apparent but it turns out, that this type of events were already studied by Dobbertin at FSE'94 under the name of normality [28]. A more general notion called k-normality was introduced and studied by Charpin cf. [8]. We discovered that a stronger result than Thm 6.0.1 in [15] holds: for every Boolean function on 6 variables either Z or $Z+1$ is annihilated by a product of 3 affine functions:

Theorem 5.2 (All Boolean functions in 6 variables are 3-normal). Given a Boolean function Z in 6 variables chosen uniformly at random the probability that it is 2-normal i.e. it has an annihilation of type

$$Z \cdot f \cdot g = 0 \text{ or } (Z+1) \cdot f \cdot g = 0$$

with two arbitrary affine factors f, g is equal to $2^{-1.66}$. Furthermore the probability that it is 3-normal i.e. it has an annihilation of type

$$Z \cdot f \cdot g \cdot h = 0 \text{ or } (Z+1) \cdot f \cdot g \cdot h = 0$$

with three arbitrary affine factors f, g, h is equal to exactly 100%.

Proof. Our property in invariant w.r.t. ordinary affine equivalence of Boolean functions w.r.t arbitrary invertible affine transformations on the 6 variables. We have examined all the 150357 classes of Boolean functions with 6 variables cf. [7], and found that 47446 classes are 2-normal and all 150357 classes are 3-normal.

6 A Simple Impossible Transition Attack of Degree 4

We first show a simple attack, which demonstrates why it is interesting to have a cycle on four polynomials $\mathcal{Q}_i$. We will then show how to annihilate one non-linear polynomial which will be sufficient to obtain an attack which works. This attack is designed for T-310 cipher, however we do NOT need to know the full specs of this block cipher. Our attack is a formal result on Boolean polynomials. In order to show it all we will need to know are two exact formulas (two Boolean polynomials) by which just **two** output bits 21 and 29 are computed in one round of encryption. In this form, the same attack could be potentially applied to any block cipher if only after renaming variables it would satisfy the same four transitions, three of which are trivial and which are shown on Fig. 7 below.

DES is a Feistel cipher operating on two branches of 32 bits each. T-310 has 4 branches with 9 bits each and bits are numbered 1..36. In one round of encryption, cf. Fig. 6, all bits numbered $1 \leq k \leq 36$ with $k \neq 0 \bmod 4$ are shifted to position $k+1$, and bits of type $4k+1$ are those freshly created. By convention $P(i) = j$ if round input vi cf. Fig. 6 is connected to bit number j in the input of the round. Similarly $D(5) = j$ means that wire $D5$ is connected to input j, except when $j = 0$ which would mean that a key bit used instead.

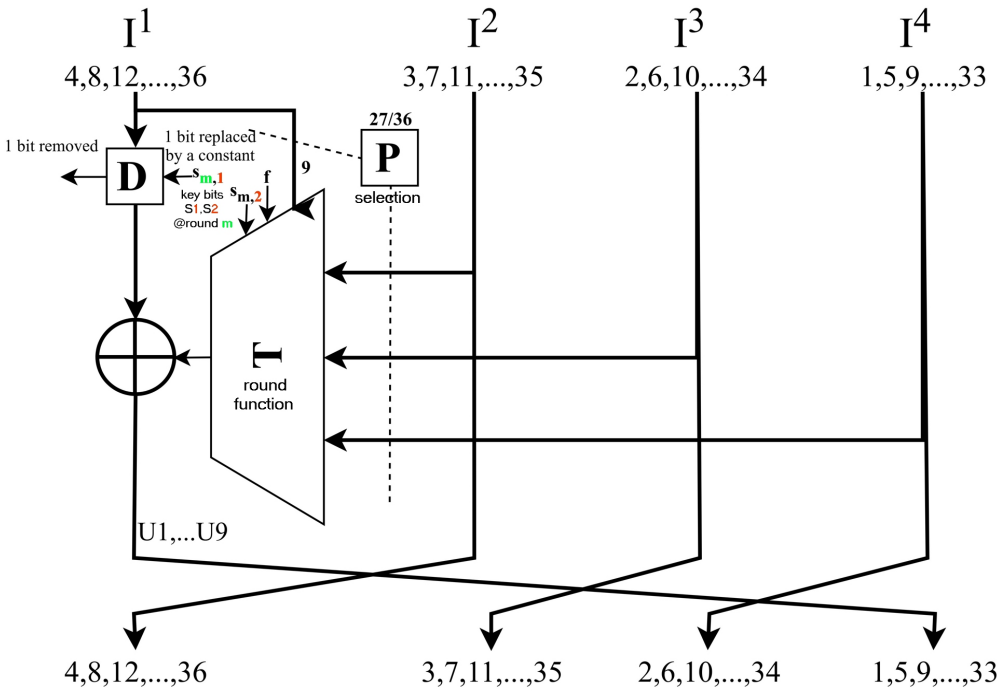

Fig. 5. T-310: a peculiar sort of Compressing Unbalanced Feistel scheme.

6.1 A Basic Nonlinear Invariant Attack of Degree 4

We will now describe an attack where the value $\in \{0, 1\}$ of a certain polynomial $\mathcal{P}$, involving only 8 bits out of 36 in each round, will be shown to be invariant after one round of T-310 block cipher.

Theorem 6.2 (Simple Invariant Attack of Degree 4). For each cipher wiring for T-310 s.t. $D(8) = P(6)$, $D(6) = 32$ and $P(10) = 30$, $P(11) = 22$ and $P(12) = 24$, if the Boolean function is such that $(Y + f)(d + e) = 0$, and for any short term key of 240 bits, and for any IV, and for any initial state on 36 bits, given the sums of 2 variables A, B, C, D defined in Fig. 7, the non-linear invariant $\mathcal{P} = ABCD$, holds with probability 1.0 for any number of rounds.

Proof. We verify on Fig. 6, that the XOR of two output bits $y_{29} + y_{21}$, is equal to a sum of 4 bits. This is exactly, following explicit general round ANF formulas given in [24, 35], equal to:

$$y_{29} + y_{21} = F + Z(...) + x_{D(8)} + F + Z(...) + x_{P(6)} + Y\left(x_{P(7)}, \ldots, x_{P(12)}\right) + x_{D(6)}$$

Following [23] cipher state variables 1-36 can also be represented as letters with a backwards numbering convention, for instance a is the same x_{36}, up to z which denotes bit x_{11}, and bits 1-10 are named by capital letters M through V. As we study 1-round invariants we need to distinguish between variables and polynomials on the input and output sides. By convention, if a represents a variable, we write a^i if it is on the input side, and a^o on the output side. Then if $D = h + p$ is a polynomial and sum of two variables, where h is also variable number 29 and p is the variable number 21, we have by definition:

$$D^i = h^i + p^i = x_{21} + x_{29}$$

where addition is modulo 2 and x_i are inputs of one round numbered from 1 to 36. In the same way if y_i are outputs of the analyzed round, by definition:

$$D^o = h^o + p^o = y_{21} + y_{29}$$

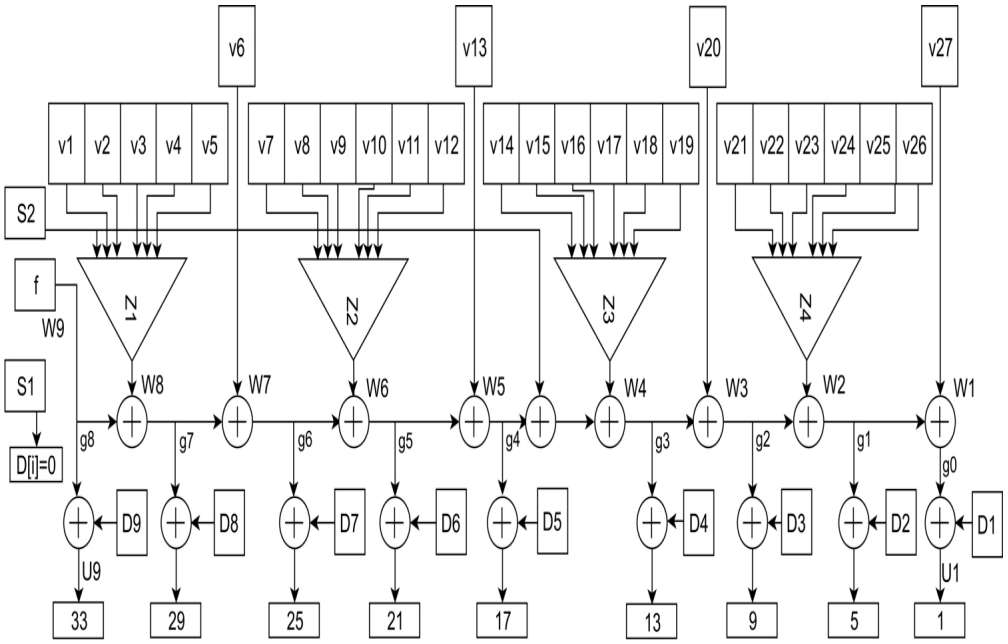

Fig. 6. The internal structure of one round of T-310 block cipher.

We can now plug-in the only formula (above), which comes from the specs of the cipher getting:

$$D^o = x_{D(8)} + x_{P(6)} + Y() + x_{D(6)}$$

Now we use our assumptions $D(8) = P(6)$, and $P(12) = 24$ and we get:

$$D^o = Y\left(x_{P(7)}, \ldots, x_{P(12)}\right) + x_{D(6)}$$

Now given that $D(6) = 32$ and $A^i = x_{24} + x_{32}$ we have:

$$D^o = Y\left(x_{P(7)}, \ldots, x_{P(12)}\right) + x_{24} + A^i$$

where x_{24} is the same as last input f of this Boolean function $Y()$, which is due to $P(12) = 24$. If for simplicity we denote by $(Y + f)$ a modified Boolean function with addition of the last input, we obtain:

$$D^o = (Y + f)\left(x_{P(7)}, \ldots, x_{P(12)}\right) + A^i$$

In addition we have the trivial transitions

$$C^o = y_{30} + y_{22} = x_{29} + x_{21} = D^i$$

and similarly $B^o = C^i$ and $A^o = B^i$, which comes from bit k becoming $k + 1$ in the next round for all k not being a multiple of 4, cf. [24, 35]. Here we will see why forming **a cycle** with A, B, C, D matters.

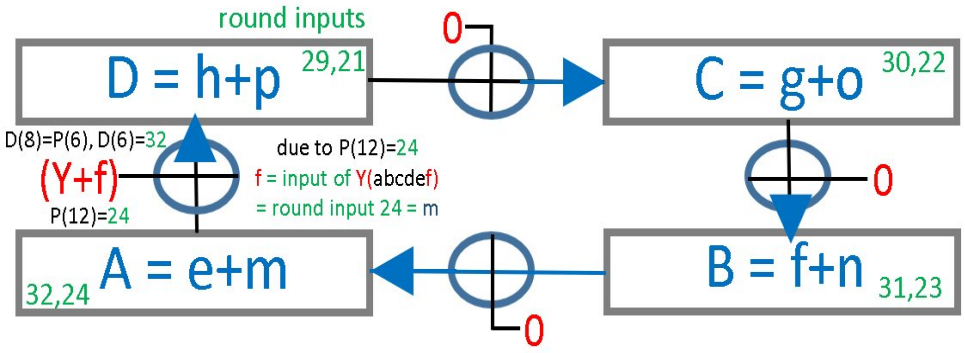

Fig. 7. A cycle and sequence of 4 polynomial transitions for 4 rounds which leads to a non-linear invariant attack with $\mathcal{P} = ABCD$, which is an invariant for 1 round.

In order to show that $ABCD$ is an invariant we need to show that:

$$\mathcal{P}^o = A^o B^o C^o D^o = A^i B^i C^i D^i = \mathcal{P}^i$$

and we have several immediate trivial transitions:

$$\mathcal{P}^o = A^o B^o C^o D^o = B^i C^i D^i D^o =$$

$$B^i C^i D^i \left((Y + f)\left(x_{P(7)}, \ldots, x_{P(12)}\right) + A^i\right) \stackrel{?}{=} A^i B^i C^i D^i = \mathcal{P}^i$$

All we need to do now is to show that:

$$B^i C^i D^i \left((Y + f)\left(x_{P(7)}, \ldots, x_{P(12)}\right)\right) = 0$$

Finally we also check that $(Y + x_{24})C^i = 0$ is the same as $(Z + f)(d + e) = 0$ due to $C^i = x_{30} + x_{22}$ and $P(10) = 30$, $P(11) = 22$ and $P(12) = 24$. □

6.3 Why Current Attack is Unsatisfactory

It is easy to note that in the last example we have NOT used the full power of the attack. We have requested that $(Y + f)C = 0$ in order to make sure that $(Y + f)BCD = 0$, in other words we must make sure that $Y + f$ is equal to zero in 2^5 points, while we would only need $Y = f$ at 2^3 entries in the truth table. This is due to the fact, that the wiring of the cipher satisfies an extremely complex set of technical requirements known under the name of KT1 specification cf. [24, 35]. These requirements seems to prevent our attack in some way. It is clearly meant to prevent attacks where several linear polynomials B, C, D would simultaneously be composed of too many inputs of the same Boolean function Y. We discover that the cryptologists in the former Eastern Bloc have somewhat managed to make our attack harder to apply.

In addition, it is difficult to hope that $(Y + f)C = 0$ could be true for any Boolean function resembling those found in real-life ciphers. Extremely few Boolean functions have linear annihilators. One basic result is as follows:

Theorem 6.4 (Impossibility for Balanced-ness and Non-Linearity). It is not possible to find a Boolean function Z required by our degree 4 attack of Thm. 6.2 in such a way, that $Y + f$ is simultaneously balanced and non-linear.

Proof. Let:

$$\mathrm{g(a, b, c, d, e, f) * Z(a, b, c, d, e, f) = 0}$$

where $g()$ is an affine function. Since g is balanced for some 2^{6-1} inputs we have $g = 1$ and for all those we must have $Z = 0$. Now since Z is balanced it must be $Z = 1$ on all the remaining 2^{6-1} inputs and our function is now completely determined and we have $Z = g + 1$ for any input. Finally since g is linear Z also must be linear. This contradiction ends our proof. □

Observations. With this result, in theory $Y + f$ would not be balanced but Y could after all be balanced. However most ciphers not only use balanced Boolean functions, but also avoid many other correlations carefully, and most modified functions such as $Y + f$ should be balanced or very close to balanced.

6.5 Discussion and Way Forward

We would like to be able design better invariant attacks which operate under weaker assumptions, such as for example $Z(a + b)c(1 + e) = 0$. This sort of annihilation properties with products of 3 affine functions do happen for real-life cryptographic Boolean functions and we have already seen cf. Thm. 5.2 that annihilations with 3 linear factors are in general totally impossible to avoid.

Until now polynomial invariant attacks seem to occur by some sort of coincidence. For example in one very special setting known in [18] as LZS 265 we require that our Boolean function satisfies $Z(a + b)(c + d)(e + f) = 0$, which was not[20] the case for the original Boolean function. We are looking towards

[20] However it is sufficient to modify just the last linear term in order to make the attack work in T-310, cf. Section 7.2. in [18].

constructing a broader family of attacks, which require a larger variety of annihilation conditions. In this paper we show how to construct such attacks in general and potentially for any block cipher. We start by stating our general construction which clearly generalizes the current attack and then we will apply it to construct a new attack on T-310 of degree 8. Moreover it should be obvious that more and better attacks can be obtained at higher degrees.

7 Our General Framework Theorem and Construction

Here is our general attack where we multiply polynomials over one or several cycles. It is possible to distinguish two sorts of transitions in our directed graphs. Simple transitions, where the difference is zero[21] and the input polynomial $\mathcal{Q}_i$ will be called a "transformable"[22] polynomial. Then we have the "impossible transitions", where the difference is a sum of complex non-linear polynomials[23].

Theorem 7.1 (General Cycling Product Invariant Attack). We consider a set of basic polynomials $\mathcal{Q}_j$ organised in one or several closed loops (directed cycles). Let $\pi(j)$ be the next point on any given cycle where $\pi()$ is an arbitrary permutation (which acts on a union of one or several directed cycles and advances one step forward). We assume that for any j we have (due to internal connections inside the cipher) the following simple transition with a XOR with a non-linear function:

$$\mathcal{Q}^o_{\pi(j)} = \mathcal{Q}^i_j + \mathcal{Z}_j()$$

where $\mathcal{Z}_j()$ are some arbitrary non-linear polynomials[24] using arbitrary variables (which represent some bits inside the cipher). Then we assume that at least **one** of these polynomials is equal to 0, i.e. we have just one trivial transition without any extra non-linear terms. Among all these polynomials those $\mathcal{Q}_j$ inside "simple" transitions as defined above, i.e. exactly those where $\mathcal{Z}_j() = 0$, are called "transformable" polynomials[25]. Other polynomials $\mathcal{Z}_i$ are non-zero[26] and they use the same set of input-side variables as $\mathcal{Q}_i$. Moreover we assume that for every $\mathcal{Z}_j()$ this Boolean function is annihilated by product of (up to) all "transformable" polynomials $\mathcal{Q}_k$, or more precisely that:

$$\forall_j \qquad \prod_{\substack{k \\ \text{transformable}}} \mathcal{Q}_k() \cdot \mathcal{Z}_j() = 0$$

[21] We have 0 in red which is XORed at three places in Fig. 7.

[22] This name means that our block cipher transforms it into another polynomial $\mathcal{Q}_j$ included in our set.

[23] For example $\mathcal{Z}_1 = Y + f$ is XORed at one place in Fig. 7 where Y is a polynomial with 6 inputs.

[24] These polynomials appear in red on our pictures for example $(Y + e)$ where Y is an arbitrary polynomial and e is an additional variable.

[25] Typically about half of all polynomials are "transformable" in all known applications of this theorem.

[26] These polynomials are exactly the same as the notion of *Transition Equation* or (TE) which was introduced in Section 5 of [17] to extend the concept of Fundamental Equation (FE) of [17] to arbitrary transitions of type $\mathcal{P} \to \mathcal{Q}$ when $\mathcal{P} \neq \mathcal{Q}$.

for any input. Then

$$\mathcal{P} = \prod_j \mathcal{Q}_j$$

is an invariant for our cipher holding with probability 1, for any secret key, for any initial state on n bits, and for any number of rounds.

Remark. Our attack is non-trivial, if this final product $\mathcal{P} \neq 0$. This needs to be checked in each case, and when $\mathcal{P} = 0$ our attack fails.

Proof. This theorem is formulated in such a way, that the proof is extremely simple. We have

$$\mathcal{P}^o = \prod_j \mathcal{Q}_j^o = \prod_j \mathcal{Q}_{\pi(j)}^o = \prod_j \left(\mathcal{Q}_j^i + \mathcal{Z}_j()\right) =$$

and now the product of all transformable polynomials can be put aside as a factor:

$$= \prod_{\substack{j \\ \text{transformable}}} \mathcal{Q}_j^i() \cdot \prod_{\substack{j \text{ not} \\ \text{transformable}}} \left(\mathcal{Q}_j^i + \mathcal{Z}_j()\right) =$$

and here each and every non-zero $\mathcal{Z}_j()$ is annihilated because the product of all transformable polynomials is a factor, and because our theorem assumes these annihilations. We get:

$$= \prod_j \mathcal{Q}_j^i = \mathcal{P}^i \quad \square$$

Observations. This theorem somewhat implicitly assumes quite a few polynomials $\mathcal{Z}_j()$ are equal to zero, which increases the number of "transformable" factors and thus in turn increases the chances for different non-zero polynomials $\mathcal{Z}_j()$ to be annihilated. It is easy to see that older Thm. 6.2. is a direct application of new Thm. 7.1., where all of the B, C, D are transformable polynomials, one of which was actually used to annihilate $Y + f$. We claim (and intend to show) that this theorem can be used to construct a large variety of attacks on block ciphers and that results compare favourably to other known attacks. This is due to the fact (as explained in Section 4.2) that the more polynomials we multiply, the easier it is to obtain the annihilations we need. In addition we enjoy a substantial freedom in the choice of these polynomials for any given block cipher. The attacker is happy to observe that there exist vast number of possible choices of the $\{\mathcal{Q}_i; \mathcal{Z}_i\}$ some of which might lead to a working attack.

8 Application to T-310: A Better Cycling Attack

We present an improved attack on T-310 which is of degree 8 and[27] is a direct application of Thm. 7.1. The application is shown on Fig. 8 below, the $\mathcal{Q}_i$ are 8 polynomials on the edges of our cycle, out of which D, C, B, H, G, F are "transformable" polynomials which we are allowed to use in our annihilation attempts

[27] A simpler example of a cycle of length 8 in T-310 is shown on Fig. 4 however the actual invariant studied was of degree 2, cf. Section 7.4 in [17].

and there are two non-zero polynomials to annihilate $W()+e$ and $Y()+e$. Given the fact that both Boolean $W()$ and $Y()$ are by definition identical, these two polynomials are annihilated in exactly the same way, modulo renaming their 6 inputs.

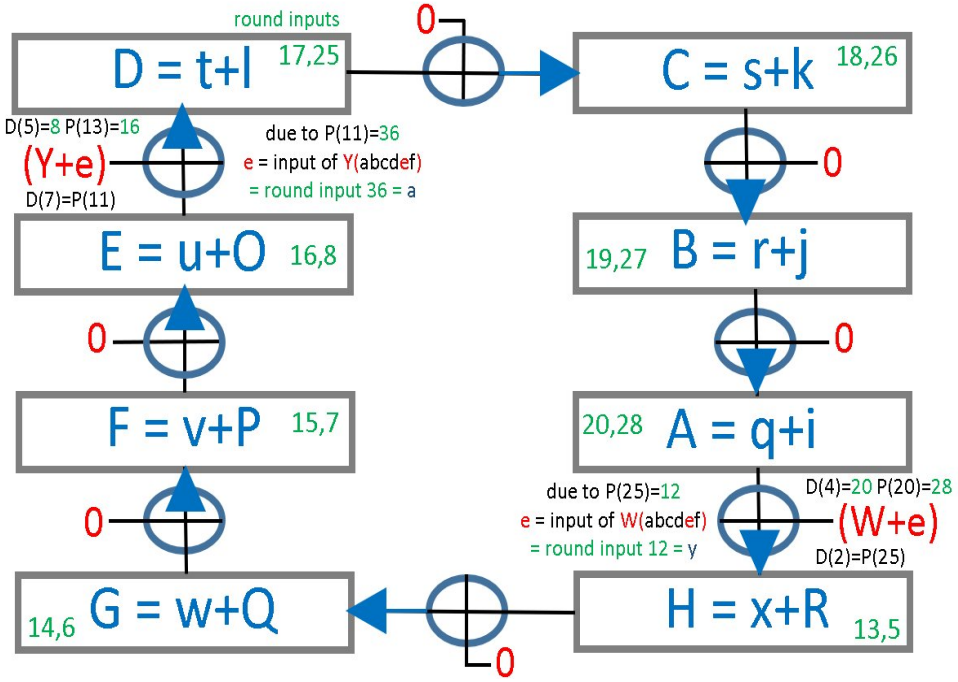

Fig. 8. An attack using a cycle on 8 with $\mathcal{P} = ABCDEFGH$.

Theorem 8.1 (An Invariant Attack of Degree 8). With polynomials $A-H$ defined as on Fig. 8, for each cipher wiring for T-310 s.t. $D(5) = 8, P(13) = 16, D(7) = P(11), D(4) = 20, P(20) = 28, D(2) = P(25)$, if the Boolean function (used twice as W and as Y for different sets of inputs) is such that

$$(Z + e)(a + b)(c + d) = 0$$

and if the first 4 inputs of W are in order bits 14,6,15,7, and the first 4 inputs of Y are in order bits 18,26,19,27 also[28], then and for any short term key of 240 bits, and for any initial state on 36 bits, we have the non-linear invariant

$$\mathcal{P} = ABCDEFGH$$

holding with probability 1.0 for any number of rounds.

Note: This is for example achieved for the following full cipher wiring:

[28] These 8 conditions are simply 8 additional conditions on $P()$ e.g. $P(22) = 14$ etc.

```
444: P=17,8,33,32,10,20,18,26,19,27,36,24,16,2,21,
34,1,25,13,28,14,6,15,7,12,23,30 D=0,12,4,20,8,32,36,16,24
```

and with a Boolean function being for example:

$$Z = fedcb + fedca + fedc + fecba + fecb + feca + fec + feba + feb + fea + fe$$

$$+fdb + fd + fcb + fc + fb + f + edcb + ed + ec + dcb + dca + da + d + cb + a + 1$$

Proof. The proof is the same as before and both follow the same principle as in Thm. 7.1 We compute

$$\mathcal{P}^o - \mathcal{P}^i =$$

and obtain the following difference, where B, C, D, F, G, H are all transformable polynomials cf. Fig. 8:

$$BCDFGH \cdot ((E + a + Y)(A + y + W) + AE)$$

A quick analysis discarding factors such as D which have variables not used as inputs of out Boolean functions W, Y and renaming variables shows that we need that $FG(y + W) = 0$ and $BC(Y + a) = 0$. Each of 2 terms is cancelled through two identical annihilation requirements of type exactly:

$$(Z + e)(a + b)(c + d) = 0$$

which is exactly our assumption.□

Remark. With invariants of degree 8 one can do better than $(Z + e)(a + b)(c + d) = 0$. In [18] we discover that it is possible to design a similar attack with a weaker assumption $Z(a + b)(c + d)(e + f) = 0$ which makes that the attack is substantially more likely to work, when the Boolean function is chosen at random. Equivalently, our attack should require to check or modify only a small number of values inside the truth table of this Boolean function Z.

9 On Existence of Polynomial Invariants in DES

DES is one of the most widely used cryptographic algorithms of all times and there exists numerous modified versions of DES [26, 33, 34]. There are strong connections between various Feistel ciphers used in government communications during the Cold War cf. [10, 26]. In Eastern Germany DES was implemented inside a portable electronic cipher machine T-316 [26]. Our methodology and notations emphasise similarities between different ciphers, and we do not believe that a cipher must necessarily be special or weak, in order to exhibit a large variety of polynomial invariants. We will now show that our attack and our systematic construction applies to DES, and allows one to build invariants true with probability 1 for a fraction of the key space. This was never done before and should be seen as improved bi-linear attack on DES of Crypto 2004, cf. [19]. A key observation is that we have phase transition: if we increase the degree of our

invariant polynomial, the probabilities that our invariant works can be improved very substantially. Again, we formulate the attack in such a way that the full description of DES is not needed. We just need to see that DES happens to satisfy a number of internal wiring conditions (full detailed wiring is shown later in Fig. 12 28) which after renaming variables could hold for any other cipher. By convention we assume that the secret key of DES is part of the S-box, or is added inside each S-box, i.e. we consider each S-box as a function of variables named $abcdef$ and not those named $ABCDEF$ cf. Fig. 9. This, as we will see later, allows our invariant attacks to be formulated in a surprisingly simple and compact way without reference to the secret key, which will be dealt with later, when we want to apply such results in cryptanalysis, cf. Remark 2 in page 22. Our specific attack below is not meant to work for real-life DES S-boxes. It will work only if the S-boxes (including the key) satisfy some specific annihilation conditions. These are stronger than in any previous non-linear invariant attack on DES [18, 19]. In Appendix B we tentatively consider what happens with real-life S-boxes.

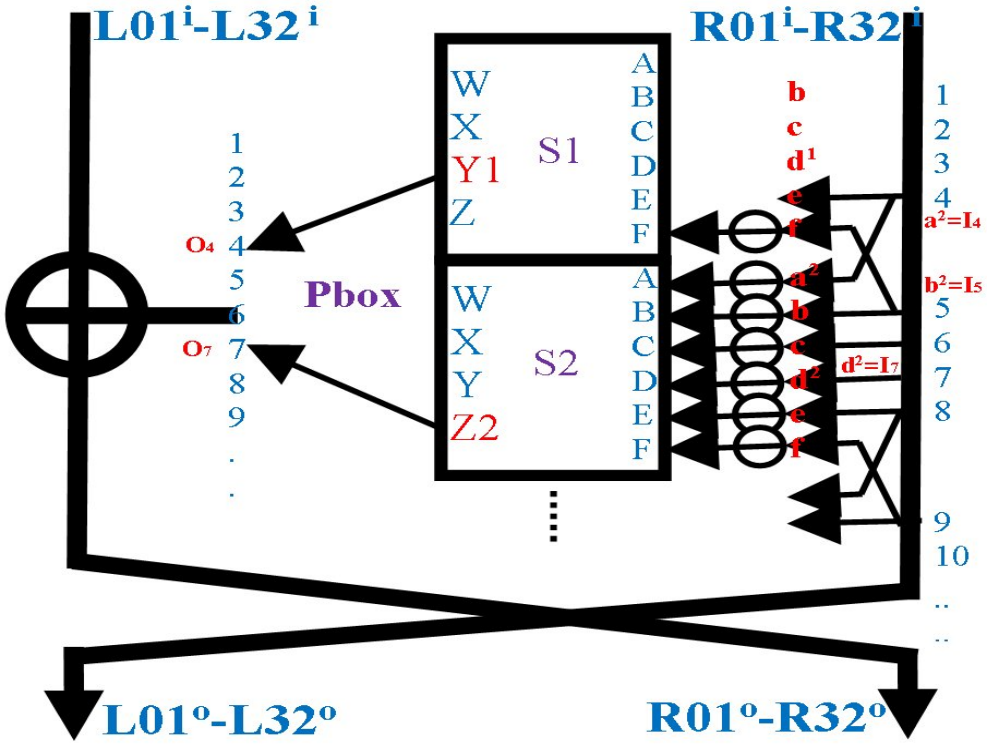

Fig. 9. One round of DES with specific notations we use in this paper.

Notation: We denote by $(\mathrm{L01},\ldots,\mathrm{L32};\mathrm{R01},\ldots,\mathrm{R32})$ the inputs of one rounds of DES. The same notations will be also used for the outputs and when it is needed to distinguish between different instances of the same variable we

will use exponents, for example $\mathrm{L05}^i$ will be the 5-th input in one round and $\mathrm{L05}^o$ will be the 5-th output bit. Now we define the following polynomials:

$$\begin{cases} A = \mathrm{R05} \in \{\text{Input bits of } \mathbf{S2}\} \\ B = \mathrm{R07} \in \{\text{Input bits of } \mathbf{S2}\} \\ C = \mathrm{R28} + 1 \in \{\text{Input bits of } \mathbf{S7}\} \cap \{\text{Input bits of } \mathbf{S8}\} \\ D = \mathrm{R27} + 1 \in \{\text{Input bits of } \mathbf{S7}\} \\ E = \mathrm{R32} \in \{\text{Input bits of } \mathbf{S8}.\} \end{cases}$$

Moreover we also define

$$\begin{cases} A' = L05, & B' = L07 \quad C' = L28 + 1 \\ D' = L27 + 1 & E' = L32 \end{cases}$$

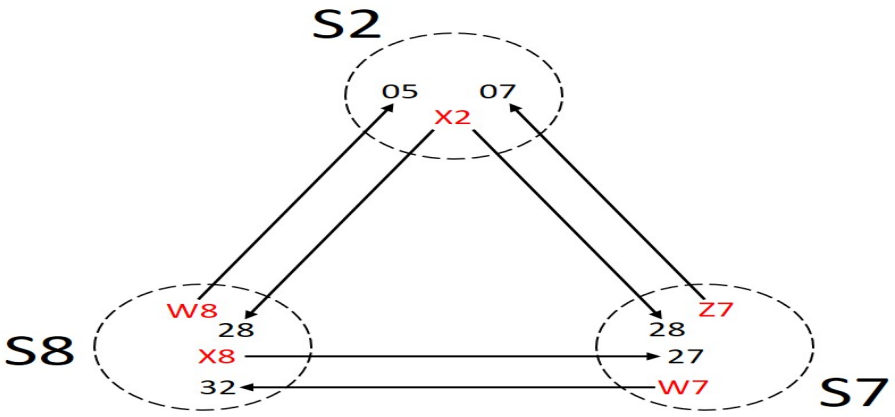

Fig. 10. Closed-loop connection between S-boxes S2,S7,S8 in DES.

We will then write that

$$(a + e) * (e) * \mathrm{W8} == 0$$

if and when the polynomial $(a + e) * (e)$ annihilates the 1st output W of the eighth S-box $S8$, where $a - f$ are inputs of that same respective S-box. By annihilation we mean that the product is zero for any input on 6 bits, i.e. $==$ is formal equality of two polynomials. At other places it there is no ambiguity we will write simply $=$ instead of $==$. We now apply our framework of Thm. 7.1 and construct our attack on DES. Our $\mathcal{Q}_i$ will be some well chosen polynomials such as C or E defined above , with for example:

$$\mathrm{W8} * C * E = 0$$

Theorem 9.1 (A Simple Degree 10 Attack On DES). We assume that:

$$\begin{cases} (a+e)*(e)*\mathrm{W8} == 0 \\ (a+e)*(e)*\mathrm{X8} == 0 \\ (d+1)*(e+1)*(\mathrm{Z7}+d) == 0 \\ (d+1)*(e+1)*(\mathrm{W7}+e) == 0 \\ bd*(\mathrm{X2}+b+d) == 0 \end{cases}$$

We also assume[29] the following connections inside the P-Box of DES, cf. Fig. 12:

$$\begin{cases} P(5) = 29 = \mathrm{W8} \\ P(7) = 28 = \mathrm{Z7} \\ P(28) = 6 = \mathrm{X2} \\ P(27) = 30 = \mathrm{X8} \\ P(32) = 25 = \mathrm{W7} \end{cases}$$

Then the product $\mathcal{P} = ABCDEA'B'C'D'E'$ which is equal to

$$\mathcal{P} = \mathrm{R05}*\mathrm{R07}*(\mathrm{R28}+1)*(\mathrm{R27}+1)*\mathrm{R32}*\mathrm{L05}*\mathrm{L07}*(\mathrm{L28}+1)*(\mathrm{L27}+1)*\mathrm{L32}$$

is a one-round invariant for DES for a fraction of key space.

Proof. We provide two proofs of Thm. 9.1 which are given in Appendix A.

Remark 1: Our conditions only concern 1/4 of values in truth table for certain outputs in three S-boxes S2,S7,S8 and the content of the remaining five S-boxes can be arbitrary.

Remark 2: This attack works for a fraction of key space which is frequently larger than it seems. It is easy to see that when we translate the input of any of our Boolean function by a secret key (by a bitwise XOR), each of our properties such as $(\mathrm{a}+\mathrm{e})(\mathrm{e})*\mathrm{W8} == 0$, still holds with probability being at least 2^{-2}. This is because each annihilating product such as $(a+e)(e)$ is itself not changed (!), with a large probability, when we modify the secret key inside the S-box.

Remark 3: As the degree of our polynomials increases, we can apply our framework and search for better attacks, such that all annihilations required have 3 affine terms. This increases the success probability that the attack works for some S-boxes. At the same time the number of active S-boxes increases. The conclusion is that there exist an optimal size where the attack is the strongest possible. Interestingly there is also a question of optimal size in truncated differential attacks [10] and both questions are related, cf. Section 3.

[29] By convention we work backwards from output to input side, cf. Fig. 12, and $P(5) = 29$ means that the output 29 of 8 S-boxes connected to round output 5, where numbering goes from 1 to 32. These connections are true for DES, and our attack works also for DES with any modified P-box for as long as it satisfies these conditions.

10 An Attack on DES with a Lower Degree of 5

We now show that the degree in our attack of Thm. 9.1 can be reduced from 10 to 5. This attack requires a stronger set of annihilations as shown below.

Theorem 10.1 (2-R invariant of degree 5 derived from degree 10 invariant). If we have the following annihilation conditions:

$$\begin{cases} \mathrm{Z7} * (e+1) = 0, & \mathrm{W7} * (e+1) = 0, \\ \mathrm{X8} * (a+1) = 0, & \mathrm{W8} * e = 0, \\ \mathrm{X2} * d = 0 \end{cases}$$

then the polynomial

$$\mathcal{P} = \mathrm{R05} * \mathrm{L07} * (\mathrm{R28}+1) * (\mathrm{L27}+1) * \mathrm{L32}$$

is an invariant after two rounds of encryption for DES.

Proof. All we need to do is to show that $\mathcal{P}^{\phi} = \mathcal{P}^{\phi^{-1}}$. We have:

$$\mathcal{P}^{\phi} = (\mathrm{L05} + \overbrace{W8_{\mathrm{R28}=\mathrm{R32},\mathrm{R01}}}^{W8*R32=0}) * \mathrm{R07} * (\mathrm{L28} + \overbrace{\mathrm{X2}_{\mathrm{R04}=\mathrm{R09}}}^{\mathrm{X2*R07=0}} + 1) * (\mathrm{R27}+1) * \mathrm{R32} =$$
$$= \mathrm{L05} * \mathrm{R07} * (\mathrm{L28}+1) * (\mathrm{R27}+1) * \mathrm{R32}$$

where the annihilations occur in the first round and R32 and R07 are inputs of the first round. For the second round we have:

$$\mathcal{P}^{\phi^{-1}} = \mathrm{L05} * (\mathrm{R07} + \overbrace{Z7_{L24=L29}}^{\mathrm{Z7*(L28+1)=0}}) * (\mathrm{L28}+1) *$$
$$(\mathrm{R27} + \overbrace{X8_{L28=L32,L01}}^{\mathrm{X8*(L28+1)=0}} + 1) * (\mathrm{R32} + \overbrace{W7_{L24=L29}}^{\mathrm{W7*(L28+1)=0}}) =$$
$$= \mathrm{L05} * \mathrm{R07} * (\mathrm{L28}+1) * (\mathrm{R27}+1) * \mathrm{R32}$$

Here the annihilations occur in the second round and if R28 is taken at the output of the second round, this variable is in fact equal to L28, or e, when seen as input of the second round S-box S7.

□

Remark 1. It may seem that this proves that the invariant attack works also for 1 round, but it doesn't. After 1 round the two sides L, R are SWAPPED[30].

Remark 2. If we denote by $\mathcal{P}'$ the symmetric version of $\mathcal{P}$ above, then it is easy to see that $\mathcal{P} + \mathcal{P}'$ is an invariant of degree 5 for 1 round, cf. Section 4.3.

Remark 3. It is an open problem if a second proof of Thm. 10.1 can at all be obtained from our general framework construction of Thm. 7.1.

[30] This is closely related to the question of reflection attacks in GOST, cf. [27].

11 Conclusion

Non-linear attacks on block ciphers are about two very complex polynomials being equal and as such so far, they were considered to happen by some extraordinary coincidence. In this paper we introduced a method for constructing non-linear attacks on block ciphers in a systematic way. It explains **why** some previously studied attacks work, and we can construct a large variety of new attacks of higher degree. We generalize the concept of closed-loop configuration cf. [38], and extend it to cycles involving arbitrary polynomials $\mathcal{Q}_i$ which cycles initially are rather impossible and are made possible by considering addition of some well chosen non-linear polynomials $\mathcal{Z}_i$. Then we eliminate all the $\mathcal{Z}_i$ algebraically. We obtain a large family of new attacks, which can now be studied.

Our general attack is formulated in such a way, that after renaming the variables it could apply to any block cipher; cf. our framework Thm. 7.1 and all our application results. The attack is built on premises that the cipher satisfies a small number of initial assumptions, which concern a tiny fraction of the specification of the cipher. This should be compared to certain results on worst-case algebraic attacks on stream ciphers. All these attacks are about how eliminating a large number of internal variables and many attacks on stream ciphers are also based on polynomial annihilation events [15]. At ICISC 2004, [16], we discover that some stream ciphers can be broken no matter what: for any Boolean function or S-box. Similarly here we observe, that as the number of polynomials in our cycles increases, our annihilation conditions with more linear factors become totally impossible to avoid, cf. Thm. 5.2. This enables us to construct attacks which work, provided that an (increasingly small) fraction of the truth table of our Boolean functions are such as requested. Overall our attacks become increasingly hard to avoid and are not necessarily prevented by numerous non-linearity requirements, which are typically studied in cryptography. We claim that our attacks are not[31] in general prevented with traditional block cipher designs methodology [20, 5, 33, 34], cf. Thm. 5.2. A serious limitation however is that for DES our attack only works for a fraction of the key space and not quite for the original S-boxes, cf. Appendix B. For T-310 attacks are substantially stronger and work for 100 % of keys and with arbitrary round constants.

The present work is not exhaustive and does NOT cover all possible invariant attacks, which in the general case form a polynomial ring. The main contribution of this paper is to show how to construct "product" attacks, where this ring is not trivial and not empty: existence of at least one invariant attack. Experience shows that then other better attacks with lower degree and more products may also exist, cf. Section 2.3 and [11, 12, 17]. For example we found that with a small modification in Thm 8.1 we can obtain a larger invariant ring with dimension up to 9, containing further invariants of degree 2,4 and 6. We demonstrate this for DES in Section 10: we show that the degree of the invariant property in Thm. 9.1 can be reduced from 10 to 5. In general it is not easy to know if any given non-linear attack can be constructed using general framework of Thm. 7.1.

[31] Except in more recent works specifically aiming at thwarting invariant attacks [3, 6].

References

1. Arnaud Bannier, Nicolas Bodin, and Eric Filiol: *Partition-Based Trapdoor Ciphers,* https://ia.cr/2016/493.
2. Joan Boyar, Magnus Find, René Peralta: *Four Measures of Nonlinearity,* In Algorithms and Complexity, CIAC 2013, LNCS 7878, pp. 61-72, Springer.
3. C. Beierle, A. Canteaut, G. Leander, Y. Rotella: *Proving resistance against invariant attacks: how to choose the round constants,* in Crypto 2017, Part II. LNCS, 10402, pp. 647–678, Springer 2017.
4. Tim Beyne: *Block Cipher Invariants as Eigenvectors of Correlation Matrices,* in Asiacrypt 2018, pp. 3-31, Springer, 2018. One version is avail. at https://eprint.iacr.org/2018/763.pdf
5. Don Coppersmith, *The development of DES, Invited Talk, Crypto'2000, August 2000.*
6. *Marco Calderini: A note on some algebraic trapdoors for block ciphers,* last revised 17 May 2018, https://arxiv.org/abs/1705.08151
7. Cagdas Calik and Meltem Sonmez Turan and Rene Peralta: *The Multiplicative Complexity of 6-variable Boolean Functions,* https://ia.cr/2018/002.pdf
8. Pascale Charpin: *Normal Boolean functions,* Journal of Complexity, vol. 20, Issues 2–3, pp 245–265, 2004.
9. Nicolas Courtois, Adi Shamir, Jacques Patarin, Alexander Klimov, *Efficient Algorithms for solving Overdefined Systems of Multivariate Polynomial Equations*, In Advances in Cryptology, Eurocrypt'2000, LNCS 1807, Springer, pp. 392-407.
10. Nicolas Courtois, Theodosis Mourouzis, Anna Grocholewska-Czurylo and Jean-Jacques Quisquater: *On Optimal Size in Truncated Differential Attacks,* In CECC 2014, post-proceedings in Studia Scientiarum Mathematicarum Hungarica, Volume 52, issue 2, pp. 246–254, 2015.
11. Nicolas T. Courtois, Aidan Patrick: *Lack of Unique Factorization as a Tool in Block Cipher Cryptanalysis,* Preprint, https://arxiv.org/abs/1905.04684 12 May 2019.
12. Nicolas T. Courtois: *Invariant Hopping Attacks on Block Ciphers,* accepted at WCC'2019, Abbaye de Saint-Jacut de la Mer, France, 31 March - 5 April 2019.
13. Nicolas T. Courtois, Marios Georgiou: *Variable elimination strategies and construction of nonlinear polynomial invariant attacks on T-310,* in Cryptologia, published online: 18 Oct 2019, At https://doi.org/10.1080/01611194.2019.1650845
14. Nicolas T. Courtois, Marios Georgiou: *Constructive Non-Linear Polynomial Cryptanalysis of a Historical Block Cipher,* At http://arxiv.org/abs/1902.02748.
15. Nicolas Courtois and Willi Meier: *Algebraic Attacks on Stream Ciphers with Linear Feedback,* Eurocrypt 2003, LNCS 2656, pp. 345–359, Springer. Extended version: www.nicolascourtois.com/toyolili.pdf.
16. Nicolas Courtois: *Algebraic Attacks on Combiners with Memory and Several Outputs,* ICISC 2004, LNCS 3506, pp. 3–20, Springer 2005. Extended version available on https://ia.cr/2003/125/.
17. Nicolas T. Courtois: *On the Existence of Non-Linear Invariants and Algebraic Polynomial Constructive Approach to Backdoors in Block Ciphers,* https://ia.cr/2018/807, last revised 27 Mar 2019.
18. Nicolas T. Courtois: *Structural Nonlinear Invariant Attacks on T-310: Attacking Arbitrary Boolean Functions*, https://ia.cr/2018/1242, revised 12 Sep 2019.
19. Nicolas Courtois: *Feistel Schemes and Bi-Linear Cryptanalysis,* in Crypto 2004, LNCS 3152, pp. 23–40, Springer, 2004.

20. Nicolas Courtois, Guilhem Castagnos and Louis Goubin: *What do DES S-boxes Say to Each Other?* Available on `https://ia.cr/2003/184/`, 2003.
21. Nicolas Courtois: *An Improved Differential Attack on Full GOST,* in "The New Codebreakers – a Festschrift for David Kahn", LNCS 9100, pp. 278-299, Springer, 2016.
22. Nicolas Courtois: *An Improved Differential Attack on Full GOST,* In Cryptology ePrint Archive, Report 2012/138. 15 March 2012, updated December 2015, `https://ia.cr/2012/138`.
23. Nicolas T. Courtois, Klaus Schmeh, Jörg Drobick, Jacques Patarin, Maria-Bristena Oprisanu, Matteo Scarlata, Om Bhallamudi: *Cryptographic Security Analysis of T-310,* Monography study on the T-310 block cipher, 132 pages, received 20 May 2017, last revised 29 June 2018, `https://ia.cr/2017/440.pdf`
24. Nicolas T. Courtois, Maria-Bristena Oprisanu: *Ciphertext-only attacks and weak long-term keys in T-310,* in Cryptologia, vol 42, iss. 4, pp. 316–336, May 2018. `http://www.tandfonline.com/doi/full/10.1080/01611194.2017.1362065`.
25. Nicolas Courtois, Maria-Bristena Oprisanu and Klaus Schmeh: *Linear cryptanalysis and block cipher design in East Germany in the 1970s,* in Cryptologia, 05 Dec 2018, `https://www.tandfonline.com/doi/abs/10.1080/01611194.2018.1483981`
26. Nicolas Courtois, Jörg Drobick and Klaus Schmeh: *Feistel ciphers in East Germany in the communist era,* In Cryptologia, vol. 42, Iss. 6, 2018, pp. 427-444.
27. Nicolas Courtois: *Algebraic Complexity Reduction and Cryptanalysis of GOST,* Monograph study on GOST cipher, 2010-2014, 224 pages, available at `https://ia.cr/2011/626`.
28. Hans Dobbertin: *Construction of bent functions and balanced Boolean functions with high nonlinearity,* in: FSE'94, LNCS 1008, Springer, Berlin, pp. 61–74, 1994.
29. C. Harpes, G. Kramer, and J. Massey: *A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-up Lemma,* Eurocrypt'95, LNCS 921, Springer, pp. 24–38.
30. Lars R. Knudsen, Matthew J. B. Robshaw: *Non-Linear Characteristics in Linear Cryptoanalysis,* Eurocrypt'96, LNCS 1070, Springer, pp. 224–236, 1996.
31. G. Leander, M.A. Abdelraheem, H. AlKhzaimi, E. Zenner,: *A cryptanalysis of PRINTcipher: The invariant subspace attack.* In Crypto 2011, LNCS 6841, pp. 206–221, 2011.
32. Richard J. Lipton, Kenneth W. Regan: book chapter entitled: *Nicolas Courtois: The Linearization Method,* in pages 259–262 inside the book: *People, Problems, and Proofs: Essays from Gödel's Lost Letter*, `https://www.springer.com/gp/book/9783642414213`, Springer, 2010.
33. Lauren De Meyer and Serge Vaudenay: *DES S-box Generator*, In Cryptologia, vol. 41, iss. 2, pp 153–171, 2017.
34. Kwangjo Kim, Sangjin Lee, Sangjoon Park, Daiki Lee: *Securing DES S-boxes against Three Robust Cryptanalysis,* In SAC'95, pp.145-157, LNCS 2595, Springer.
35. Klaus Schmeh: *The East German Encryption Machine T-310 and the Algorithm It Used,* In Cryptologia, vol. 30, iss. 3, pp. 251–257, 2006.
36. Adi Shamir: *On the security of DES,* Crypto'85, LNCS 218, Springer, pages 280-281.
37. Yosuke Todo, Gregor Leander, and Yu Sasaki: *Nonlinear invariant attack: Practical attack on full SCREAM, iSCREAM and Midori64,* In Journal of Cryptology, pp. 1–40, April 2018.
38. Yongzhuang Wei, Tao Ye, Wenling Wu, Enes Pasalic: *Generalized Nonlinear Invariant Attack and a New Design Criterion for Round Constants*, In IACR Tr. on Symm. Crypt. Vol. 2018, No. 4, pp. 62-79.

A Two Proofs of Thm. 9.1

We provide two proofs of Thm. 9.1. First proof follows our framework of Thm. 7.1 directly and uses definitions A, B, C, D, E of Section 9 and operates by multiplying all polynomials on two cycles cf. Fig. 11 and Fig. 13. The second proof just shows that the attack works directly step by step[32].

First Proof of Thm. 9.1: We show that our attack follows from Thm. 7.1 directly. The proof consists of rewriting everything with input variables for one round and checking for consistency. First we check all transitions on both cycles. Each output-side polynomial $\mathcal{Q}_{j'}$ is checked if it is always equal to the sum of the input-side polynomial $\mathcal{Q}_j$ and the $\mathcal{Z}_j$ polynomial added along the way.

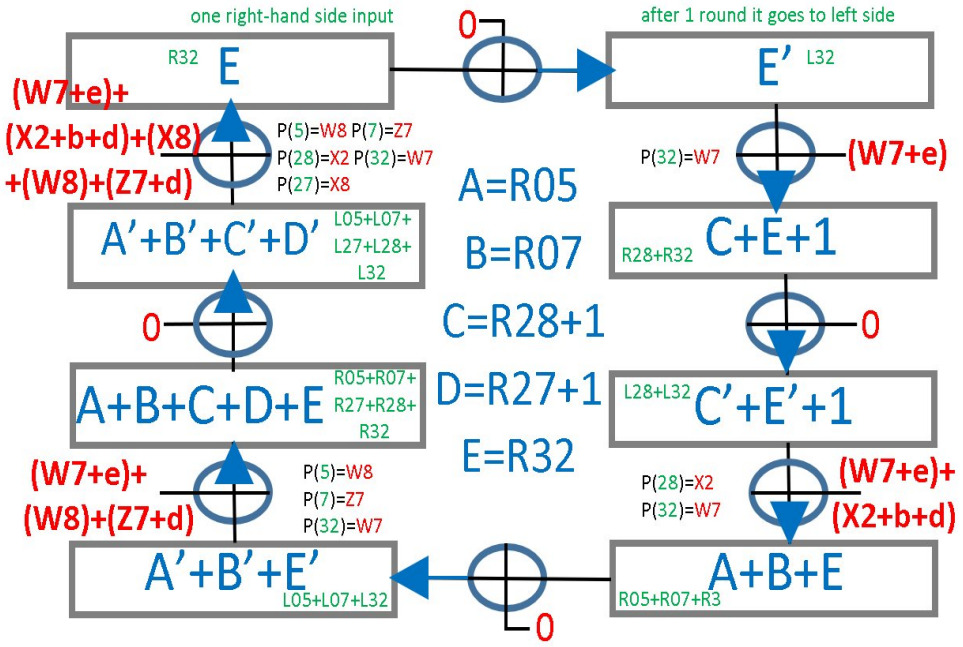

Fig. 11. First of our two cycles leading to a non-linear invariant attack of degree 10 on DES with modified S-boxes in our Thm. 9.1.

Then we show that the product of 8+8 polynomials is the same as our intended invariant $\mathcal{P}$ of degree 5+5. Here we observe that variables from left and right sides of DES never mix and that they are identical if we replace R by L. Therefore it is sufficient to check the product of every second polynomial:

$$ABCDE = \mathrm{R05} * \mathrm{R07} * (\mathrm{R28}+1) * (\mathrm{R27}+1) * \mathrm{R32} =$$
$$(\mathrm{R32}) * (\mathrm{R32}+\mathrm{R28}) * (\mathrm{R05}+\mathrm{R07}+\mathrm{R32}) * (\mathrm{R05}+\mathrm{R07}+\mathrm{R27}+\mathrm{R28}+\mathrm{R32}) *$$
$$(\mathrm{R07}+\mathrm{R28}) * (\mathrm{R05}+\mathrm{R27}+\mathrm{R28}) * (\mathrm{R07}+\mathrm{R27}+\mathrm{R28}) * (\mathrm{R05}+\mathrm{R28}).$$

[32] This without using A, B, C, D, E and without revealing that it is at all based on 2 well-chosen cycles, and that it might be an application of Thm. 7.1.

We recall that "transformable" polynomials are all $\mathcal{Q}_j$ which are transformed into another polynomial $\mathcal{Q}'_j$ included, i.e. all those with 0 added. Here they are also exactly those with letters $A, B, C, \ldots$ and not $A', B', C', \ldots$, and also exactly those with variables R01 through R32, i.e. every second $\mathcal{Q}_i$. Accordingly the identity above proves that the product of exactly **all** "transformable" polynomials on both cycles is simply equal to $ABCDE$ which fact we will use below.

Now we need to check that all the $\mathcal{Z}_j$ vanish when multiplied by $ABCDE$. We recall that on our cycles the meaning of letter a is different for each S-box, cf. Fig. 9 page 9 and also Fig. 12 below. All polynomials such as $(a + e)e$ need to be translated into round inputs.

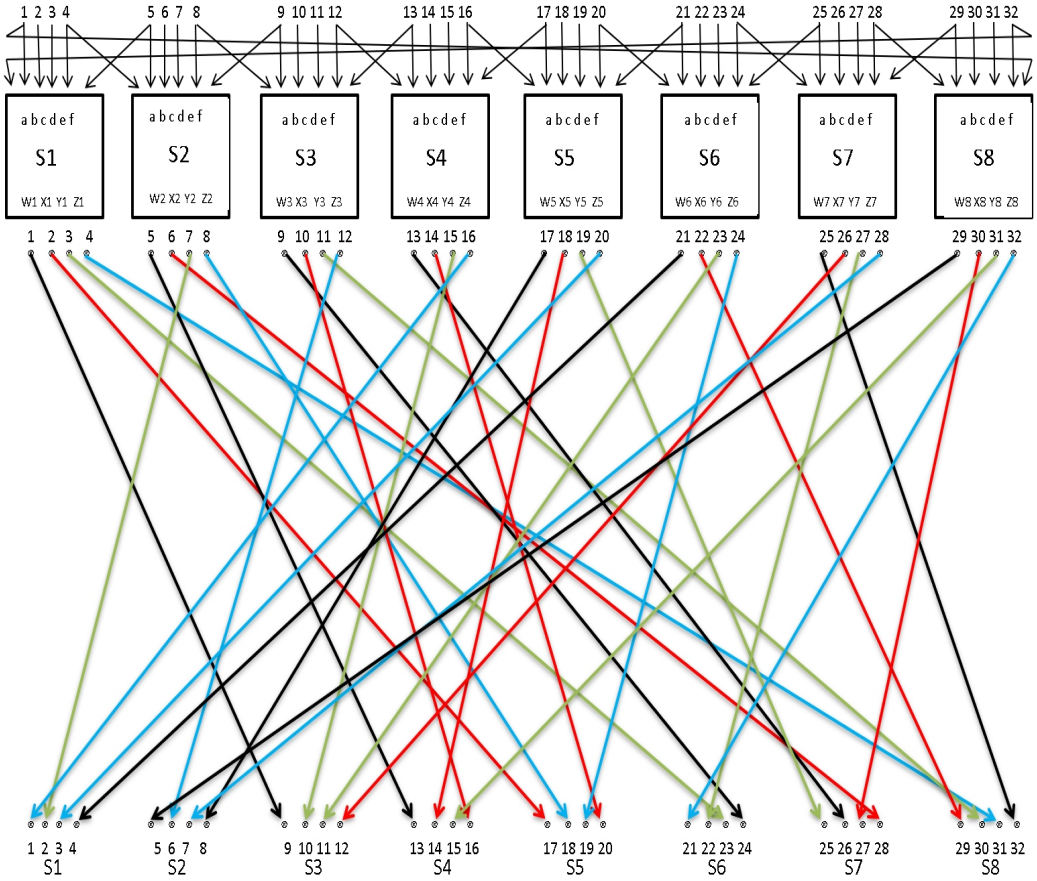

Fig. 12. Full round function of DES showing the DES P-box.

All the $\mathcal{Z}_j$ will be annihilated if we annihilate the 5 components $(W7 + e), (X2 + b + d), (X8), (W8), (Z7 + d)$. We check that each is annihilated by the product[33] of "transformable" polynomials which is equal to $ABCDE$. For

[33] However it is NOT necessary to insure that all "transformable" polynomials are used.

example we check that $W8$ is eliminated correctly: for $W8$ the polynomial $(a+e)e$ is the same as $CE = (R28+1)R32 = (R28+R32)R32$ in our attack, where $(R28+R32)$ and $R32$ are transformable polynomials present. Then the term $(\mathrm{X2}+b+d)$ is annihilated by the product of AB which is the same as $(bd) = (\mathrm{R05}\cdot\mathrm{R07})$ for S-box 2. cf. Fig. 12. Same for $X8$. For $Z7$ or $W7$ we have $(d+1)(e+1) = (R27+1)(R28+1) = CD$. We have already shown in page 27 that the product of all transformable polynomials is equal to $ABCDE$. This ends the proof.

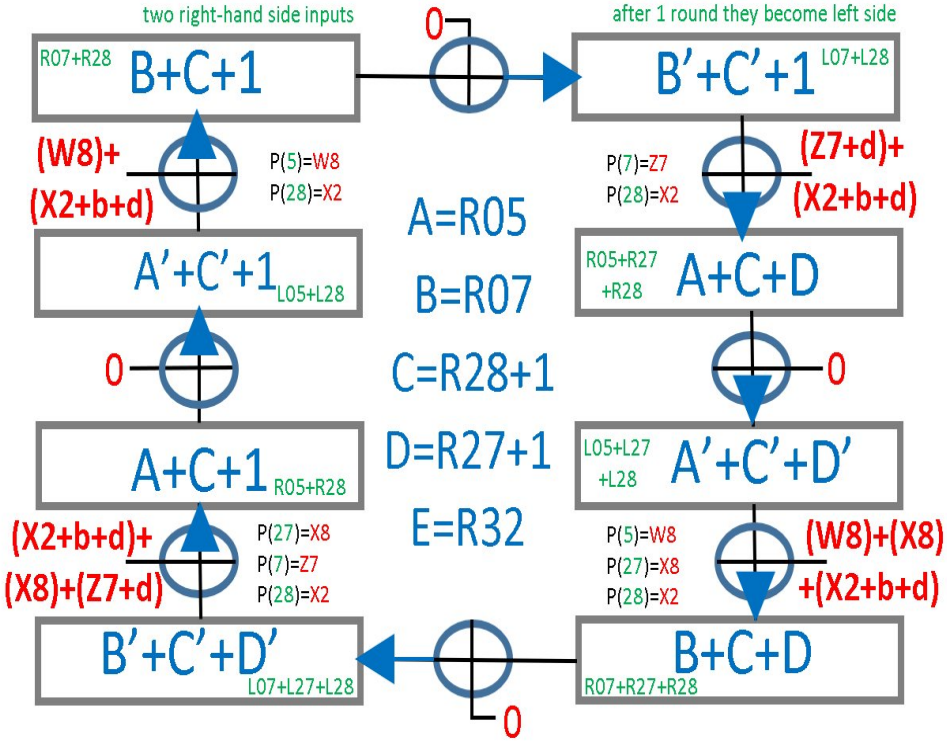

Fig. 13. Our second cycle leading to a non-linear invariant attack of degree 10 on DES with modified S-boxes in the same Thm. 9.1.

Second Proof of Thm. 9.1: We recall how our annihilation conditions are transformed:

$$\begin{cases} (a+e)*(e)*\mathrm{W8} == 0 \\ (a+e)*(e)*\mathrm{X8} == 0 \\ (d+1)*(e+1)*(\mathrm{Z7}+d) == 0 \\ (d+1)*(e+1)*(\mathrm{W7}+e) == 0 \\ bd*(\mathrm{X2}+b+d) == 0 \end{cases}$$

$$\begin{cases} \mathrm{W8}*C*E = \mathrm{X8}*C*E = 0 \\ \mathrm{Z7}*C*D = \mathrm{W7}*C*D = 0 \\ \mathrm{X2}*A*B = 0 \end{cases}$$

this will be further transformed checking which exact Boolean functions are added at the correct places cf. Fig. 12 above. After one round ϕ of encryption the polynomial $\mathcal{P}$ becomes:

$$\mathcal{P}^{\phi} = (\mathrm{L05}+\mathrm{P5})*(\mathrm{L07}+\mathrm{P7})*(\mathrm{L28}+\mathrm{P28}+1)*(\mathrm{L27}+\mathrm{P27}+1)*(\mathrm{L32}+\mathrm{P32})*$$
$$*\mathrm{R05}*\mathrm{R07}*(\mathrm{R28}+1)*(\mathrm{R27}+1)*\mathrm{R32} =$$
$$= (\mathrm{L05}+\mathrm{W8})*(\mathrm{L07}+\mathrm{Z7})*(\mathrm{L28}+\mathrm{X2}+1)*(\mathrm{L27}+\mathrm{X8}+1)*(\mathrm{L32}+\mathrm{W7})*$$
$$*\mathrm{R05}*\mathrm{R07}*(\mathrm{R28}+1)*(\mathrm{R27}+1)*\mathrm{R32} =$$

$$=(\mathrm{L05}+\overbrace{\cancel{W8}}^{\mathrm{C*E*W8=0}})*(\mathrm{L07}+\overbrace{\cancel{Z7}}^{\mathrm{C*D*Z7=0}})*(\mathrm{L28}+\overbrace{\cancel{X2}}^{\mathrm{A*B*X2=0}}+1)*(\mathrm{L27}+\overbrace{\cancel{X8}}^{\mathrm{C*E*X8=0}}+1)$$
$$*(\mathrm{L32}+\overbrace{\cancel{W7}}^{\mathrm{C*D*W7=0}})*\mathrm{A}*\mathrm{B}*\mathrm{C}*\mathrm{D}*\mathrm{E} = \mathcal{P}$$

which is exactly equal to our original polynomial $\mathcal{P}^i = \mathcal{P}$ on the input side, and thus we have proven (under our 5 annihilation assumptions) the formal equality of two polynomials, the original and the transformed one:

$$\mathcal{P}^o = \mathcal{P}^i$$

which shows that our invariant attack works for 1 round of DES. □

B Original DES Boxes: Shamir 1985 Paper Revisited

In 1985 Shamir observed that for **every** DES S-box, if we fix the second input variable to 1, the sum of all outputs is very strongly biased [36]. This has important consequences for our attacks. For **every** strongly biased Boolean function either Z or $Z+1$ has unusually many annihilators, cf. Thm. B.2. in [18]. In particular we expect to find some unusually low degree annihilators, for example, the following property holds with probability 1 for the DES S-box S5:

$$R17(R16+R20)*(W5+X5+Y5+Z5) = 0.$$

Similar results hold for other S-boxes. Furthermore, if we consider adding also some inputs, some annihilations at degree 1 also exist for DES:

$$(1+R14+R16)*(W4+X4+Y4+Z4+1+R12+R14) = 0$$
$$(1+R16+R17+R20)*(W5+X5+Y5+Z5+1+R17) = 0.$$

These two annihilators are linear which correspond to 1-weak-normality using terminology of [8]. Here potentially even the old bi-linear attack [19] applies. However most likely we need to combine these with properties of higher degree.

It is then easy to see that we are not or **not yet** using the full power of Thm. 7.1. It remains an open problem to find another application of Thm. 7.1 where the term $(W5+X5+Y5+Z5+1+R17)$ would be annihilated **directly**. Then we would need only one transformable polynomial $(1+R16+R17+R20)$. It is an open problem to see if suitable cyclic configurations of polynomials exist for DES. There are countless possibilities and possibly DES remains quite secure. This is because the strongest properties happen systematically with 4 active outputs, increasing the number of annihilations necessary in any invariant attack, and making that attacks would overall work for extremely few keys.